

## CLAIMS

What is claimed is:

1           1.     A method for emulating a plurality of virtual timers in a virtual computer  
2 system, the virtual timers being programmable by guest software to generate a plurality  
3 of timer events, the method comprising:

4                 receiving programming information from the guest software for  
5                 programming a first virtual timer;

6                 receiving programming information from the guest software for  
7                 programming a second virtual timer;

8                 determining when the first virtual timer would generate timer events if it  
9                 were implemented in a physical computer system, based on the programming  
10                information received from the guest software;

11                determining when the second virtual timer would generate timer events if it  
12                were implemented in a physical computer system, based on the programming  
13                information received from the guest software; and

14                generating timer events for the first virtual timer and the second virtual  
15                timer in the same combined sequence that they would occur if the first and  
16                second virtual timers were implemented in a physical computer system.

17  
1           2.     The method of claim 1, wherein a catch-up mode is used when the  
2 generation of timer events in the virtual computer system is behind the timing of when  
3 the timer events would be generated in a physical computer system and a normal mode  
4 is used when the generation of timer events in the virtual computer system is caught up  
5 to the timing of when the timer events would be generated in a physical computer  
6 system; wherein, when the catch-up mode is used, the average rate of timer events in  
7 the virtual computer system exceeds the average rate at which timer events would be  
8 generated in a physical computer system; and wherein, when the normal mode is used,  
9 the average rate of timer events in the virtual computer system is substantially the same  
10 as the average rate at which timer events would be generated in a physical computer  
11 system.

3. The method of claim 2, wherein, when the catch-up mode is used, the interval between successive timer events in the virtual computer system is substantially proportional to the interval that would occur between the same successive timer events in a physical computer system.

4. The method of claim 2, wherein, when the normal mode is used, the interval between successive timer events in the virtual computer system is substantially the same as the interval that would occur between the same successive timer events in a physical computer system.

5. The method of claim 2, wherein the catch-up mode is entered when the generation of timer events in the virtual computer system falls behind the timing of when the timer events would be generated in a physical computer system by a predetermined amount and the normal mode is entered when the generation of timer events in the virtual computer system goes ahead of the timing of when the timer events would be generated in a physical computer system by a predetermined amount.

6. The method of claim 2, wherein the catch-up mode is entered immediately when the generation of timer events in the virtual computer system falls behind the timing of when the timer events would be generated in a physical computer system and the normal mode is entered immediately when the generation of timer events in the virtual computer system catches up to the timing of when the timer events would be generated in a physical computer system.

7. The method of claim 1, wherein, if the guest software attempts to read a count value from a virtual timer, a count value is returned to the guest software that represents a time value that occurs after a time value that is represented by a most recent preceding timer event and before a time value that is represented by a next timer event to occur.

1           8.     The method of claim 7, wherein the time value that is represented by the  
2 count value that is returned to the guest software falls proportionately between the time  
3 value that is represented by the most recent preceding timer event and the time value  
4 that is represented by the next timer event to occur, based on the proportion at which  
5 the time of the attempted reading of the count value falls between the actual time that  
6 the most recent preceding timer event was generated and the actual time that the next  
7 timer event is scheduled to be generated.

1           9.     The method of claim 1, wherein the method is performed by keeping track  
2 of an apparent time, which represents the time as it would appear to the guest software.

1           10.    The method of claim 1, wherein the method is performed using a timer  
2 event queue.